REMARKS

The claims remaining in the present application are Claims 1-20. Claims 5, 6, 8, 17, 18, and the Specification have been amended. The drawings have been objected to and have been responded to in a submission of proposed drawing amendments filed herewith. No new matter has been added as a result of these amendments.

DRAWINGS

The drawings are objected to under 37 C.F.R. 1.84(p)(5). The reference numeral "235" has been added in several places in Figures 2A-2E. The label "C Cell" has been added in Figures 2D and 2E. The Specification has been amended at the paragraph beginning on page 24, line 18 and the paragraph beginning on page 29, line 18 to correct minor informalities. It is respectfully submitted that the amended drawings and Specification comply with 37 C.F.R. 1.84(p)(5).

CLAIM OBJECTIONS

Claims 5 and 17 stand objected. Claims 5 and 17 have been amended to clarify that they further limit d) of their respective base claims. Claims 6 and 18 have also been amended in light of the changes to Claims 5 and 17, from which they respectively depend.

CLAIM REJECTIONS

Claims 1-20 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Mason et al, U.S. Patent No. 5,946,219 (hereinafter Mason). The rejection is respectfully traversed.

Embodiments of the present invention provide for a method and system for automatically building a bit order data structure of configuration bits for a programmable logic device. An exemplary bit order data structure is shown in Figure 4C. First, an embodiment builds a configuration bit data structure from a schematic hierarchy of a programmable logic device. An exemplary configuration bit data structure is shown in Figure 1B. The configuration bit data structure contains wordline and bitline addresses and logical names for each configuration bit. Moreover, the configuration bit data structure is derived from a hierarchical schematic representation of said programmable device. Then, by using an input database that specifies the order in which the wordlines and bitlines are to be loaded, an embodiment creates the bit order data structure.

CLAIMS 1-7

Claim 1 reads:

> A computer implemented method of generating an order of loading data into a programmable device comprising the steps of:
>
> a) identifying a plurality of memory cells in a hierarchical schematic representation of said programmable device;
>
> b) automatically determining a plurality of addresses corresponding to said plurality of memory cells;
>
> c) automatically determining a plurality of logical names for said plurality of memory cells; and
>
> d) based on an order in which said plurality of addresses are to be loaded into said programmable device, automatically storing said plurality of *logical names* for said plurality of memory cells within a data structure within computer readable memory (emphasis added).

Claim 1 recites identifying a plurality of memory cells in a hierarchical schematic representation of said programmable device.

Mason fails to disclose or suggest identifying a plurality of memory cells in a hierarchical schematic representation of a programmable device, as claimed. Rather, Mason discloses a process involving the configuration of a field programmable gate array (col. 1, lines 7-8). The process in Mason involves configuring and re-configuring the FPGA. Mason describes steps that include starting with a design of one or more logic circuits that will be implemented within the FPGA (col. 2, lines 13-15). Mason discloses certain steps taken with respect to this user design. For example, in Figures 4 and 5 Mason discloses entering an initial design (that is to be implemented within the FPGA, as opposed to a design of the FPGA itself). Certain steps may be taken with respect to the user design, which is to be implemented within the FPGA. The initial user design may then be modified by the user, such that the FPGA may be reconfigured (Abstract). However, Applicants do not understand Mason to disclose identifying a plurality of memory cells in a hierarchical schematic *representation of a programmable device*, as claimed.

Moreover, Mason actually teaches away from the limitation of identifying a plurality of memory cells in a *hierarchical schematic representation of a programmable device.* Mason at col. 2 lines 57-62 and Figure 1 discloses a coordinate system to identify the location of logic cells. The cells are numbered left to right and from top to bottom in an array-like fashion, as opposed to a hierarchical

fashion. Furthermore, Applicants do not understand Mason to teach a hierarchical system elsewhere in the disclosure.

Claim 1 further recites automatically storing a plurality of logical names for a plurality of memory cells. As just discussed, Claim 1 also recites that the plurality of memory cells are in a hierarchical schematic representation of a programmable device.

Mason fails to disclose or suggest automatically storing a plurality of logical names for a plurality of memory cells in a hierarchical schematic representation of a programmable device, as claimed. In contrast, Mason describes generating a configuration bitstream from a design database, using a bitstream compiler (col. 6, lines 23-33). In this fashion, the FPGA may be programmed to implement the user design. Applicants understand the configuration bitstream to comprise a series of "ones" and "zeroes" that are fed into the FPGA. In contrast, Claim 1 recites producing a file comprising logical names. Moreover, the logical names are associated with the hierarchical schematic representation of a programmable device.

Claim 1 further recites that a plurality of addresses and logical names are automatically determined for the plurality of memory cells. Bearing in mind the earlier discussed limitations, Claim 1 recites automatically determining information from a hierarchical schematic representation of a programmable device.

In contrast, Mason fails to disclose or suggest automatically generating the recited limitations from a *hierarchical schematic representation of a programmable device*. For example, Mason discloses that a design entry module, such as a CAD tool, a schematic capture program, or the like is used to create and store an initial design. However, to create a design database from the initial design, Mason discloses that placing-and-routing is performed. "Next, is the placement and routing of the logic gates of the logic circuit, step 212. As a result of the placement and routing step, a design database is generated, step 214" (col. 2, lines 50-52). Applicants understand this process to be for a user design to be implemented in the FPGA.

For the foregoing rationale, it is respectfully submitted that Claim 1 is not taught or suggested by Mason. Thus, allowance of Claim 1 is earnestly solicited.

Claims 2-7 depend from Claim 1, which is believed to be allowable. As such, Claims 2-7 are believed to be allowable.

CLAIMS 8-13

Amended Claim 8 reads, in part:

> a) accessing a data structure comprising a plurality of logical names corresponding to a plurality of addresses;
> b) accessing a data structure specifying an order in which said plurality of addresses are to be loaded into said programmable logic device;
> c) ordering said plurality of logical names from step a) based on the order specified in said data structure in step b); and
> d) storing said ordered plurality of logical names from step c) in a data structure within computer readable memory, wherein said ordered plurality of logical names describe an order of loading data into a programmable logic device (emphasis added).

Claim 8 recites a limitation of ordering said plurality of logical names. Claim 8 also recites storing the logical names, wherein said ordered plurality of logical names describe an order of loading data into a programmable logic device.

Mason fails to disclose or suggest ordering a plurality of logical names and storing the logical names, wherein the ordered plurality of logical names describe an order of loading data into a programmable logic device. In contrast, Mason describes generating the actual data to load into the FPGA. For example, Mason describes generating a configuration bitstream from a design database, using a bitstream compiler (col. 6, lines 23-33). In this fashion, the FPGA may be programmed to implement the user design. Applicants understand the configuration bitstream to comprise a series of "ones" and "zeroes" that are fed into the FPGA. In contrast, Claim 8 recites producing a file comprising logical names. Moreover, the logical names describe an order of loading data into a programmable logic device. However, the logical names are not necessarily the data that is loaded into programmable logic device in order to program it.

For the foregoing rationale, Mason does not teach or suggest the claimed limitations. Therefore, allowance of Claim 8 is respectfully requested.

Claims 9-13 depend from Claim 8, respectively. As Claim 8 is respectfully believed to be allowable, allowance of Claims 9-13 is respectfully solicited.

CLAIMS 14-20

Claim 14 recites, in part:

> a) identifying a plurality of memory cells in a hierarchical schematic representation of said programmable device;
> b) automatically determining a plurality of addresses corresponding to said plurality of memory cells;
> c) automatically determining a plurality of logical names for said plurality of memory cells; and
> d) based on an order in which said plurality of addresses are to be loaded into said programmable logic device, automatically storing said plurality of logical names for said plurality of memory cells within a data structure within computer readable memory.

For the reasons discussed with respect to Claim 1, Mason does not teach or suggest the limitations of Claim 14. Therefore, allowance of Claims 14 is respectfully requested.

Claims 15-20 depend from Claim 14, respectively. As Claim 14 is respectfully believed to be allowable, allowance of Claims 15-20 is respectfully solicited.

Applicants have reviewed the cited references including Sakashita, U.S. Patent No 5,384,275; Elayda, U.S. Patent No. 6,438,738; and Bair et al., U.S. Patent No. 5, 278, 769. The Examiner's comments thereto are respectfully traversed. Applicants respectfully submit that Claims 1, 8, and 14 are not taught or suggested by the cited references, alone or in combination.

The Examiner states that Sakashita discloses an arranging and wiring step that is similar to "the ordering function" disclosed by the Applicants. Applicants respectfully request that the Examiner provide specific reference to claim limitations

Respectfully submitted,
WAGNER, MURABITO & HAO LLP

Dated: 12/9, 2002

Ronald M. Pomerenke
Registration No. 43,009

Address: WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone: (408) 938-9060 Voice
(408) 938-9069 Facsimile

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

Please replace the paragraph beginning on page 24, line 18 with the following rewritten paragraph:

-- If the determination in step 400 determines that the instance did not represent a logical unit other than a configuration bit 235, then the process continues in Figure [3D] 2D. In step 450, the process 300 begins a series of steps for each configuration bit 235 in the instance. In step 455, the process renumbers the configuration bit 235 as needed. --

Please replace the paragraph beginning on page 29, line 18 with the following rewritten paragraph:

-- Figure 4B illustrates an exemplary configuration block order database 208. The database 208 describes the order in which the configuration blocks are included in the bitstream used to load the configuration bits into a CPLD. Each line represents a configuration block with a unique name, for example, a top level logical name. The database 208 may also contain the logical unit type. The abbreviations in this database are defined [in Figure 4] herein in connection with an exemplary list of logical unit names for various cells and their logical hierarchy, for example, 'cl' refers to cluster. --

IN THE CLAIMS

Please amend Claims 5, 6, 8, 17, and 18 as follows:

5. The method of Claim 1 wherein said d) comprises [further comprising the step of]:

[e)] determining whether there is a configuration bit at an [said] address of said plurality of addresses in a [said] configuration block.

6. The method of Claim 5 wherein said d) further comprises [further comprising the step of]:

[f)] placing a spacer in said data structure of said plurality of logical names in [said step d)] responsive to a determination [of step e)] that there was no configuration bit at said address in said configuration block.

8. (Amended) A computer implemented method of generating an order of loading data into a programmable logic device comprising the steps of:

a) accessing a data structure comprising a plurality of logical names corresponding to a plurality of addresses;

b) accessing a data structure specifying an order in which said plurality of addresses are to be loaded into said programmable logic device;

c) ordering said plurality of logical names from step a) based on the order specified in said data structure in step b); and

d) storing said ordered plurality of logical names from step c) in a data structure within computer readable memory, wherein said ordered plurality of logical names describe an order of loading data into a programmable logic device.

17. (Amended) The method of Claim 14 wherein said d) comprises [further comprising the step of]:

[e)] determining whether there is a configuration bit at an [said] address of said plurality of addresses in a [said] configuration block.

18. (Amended) The method of Claim 17 wherein said d) further comprises [further comprising the step of]:

[f)] placing a spacer in said data structure of said plurality of logical names [in said step d)] responsive to a determination [of step e)] that there was no configuration bit at said address in said configuration block.

when comparing subject matter "disclosed" by the Applicants to prior art. Applicants are unclear as to which, if any, claim limitations are being referred to. Sakashita is concerned with manufacturing an FPGA in reduced time (Abstract). The cited passage (col. 2, lines 60-65) discloses developing logic function specifying data. Assuming the Examiner is referring to d) of Claims 1 and 14 and/or c) of Claim 8, Applicants do not understand the cited passage to disclose or suggest "the ordering function."

CONCLUSION

In light of the above listed amendments and remarks, reconsideration of the rejected Claims is requested. Attached hereto is a sheet entitled, "VERSION WITH MARKINGS TO SHOW CHANGES MADE."

Based on the arguments and amendments presented above, it is respectfully submitted that Claims 1-20 overcome the rejections of record. Therefore, allowance of Claims 1-20 is earnestly solicited.

Should the Examiner have a question regarding the instant amendment and remarks, the Applicants invites the Examiner to contact the Applicants' undersigned representative at the below listed telephone number.

OIPE JC102
DEC 17 2002
PATENT & TRADEMARK OFFICE

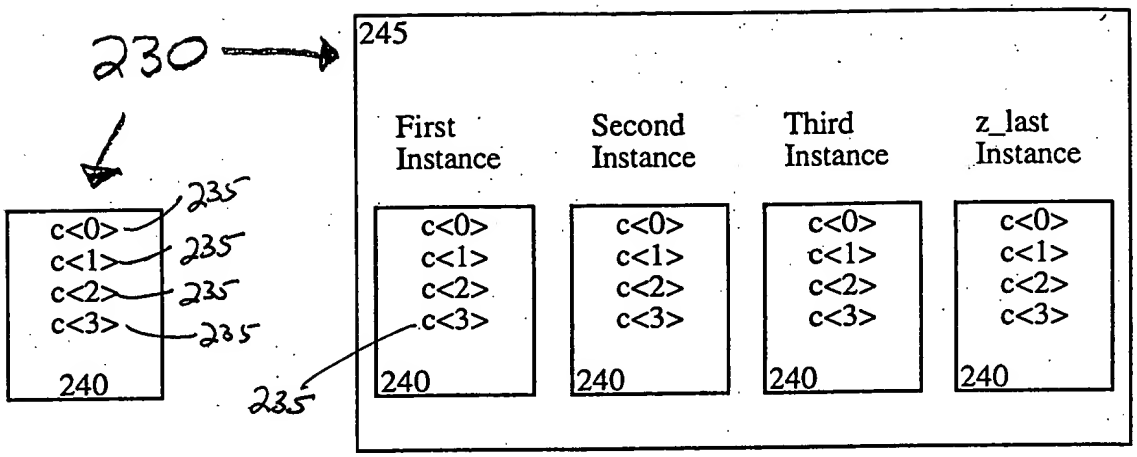

FIGURE 2A

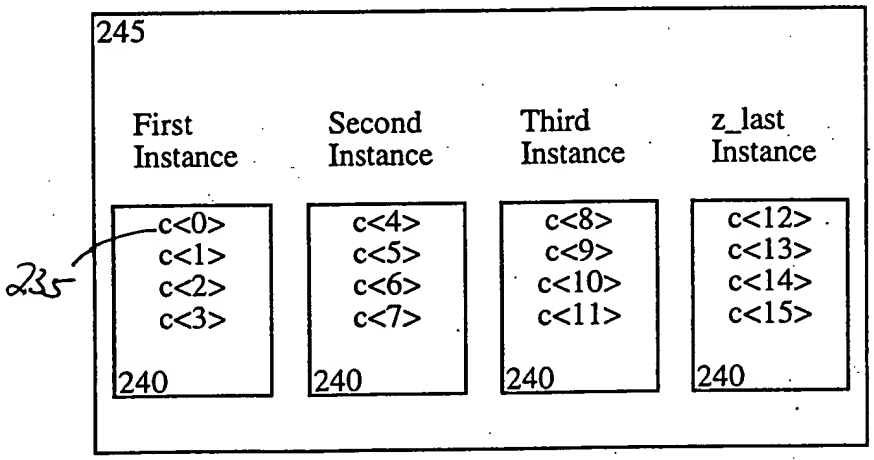

FIGURE 2B

OIPE JC102
DEC 17 2002
PATENT & TRADEMARK OFFICE

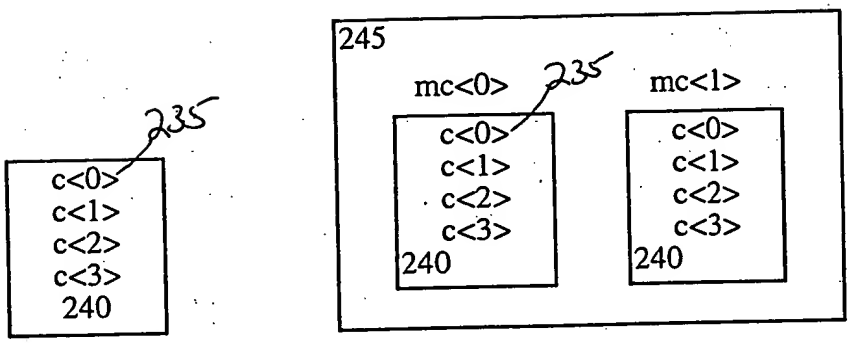

FIGURE 2C

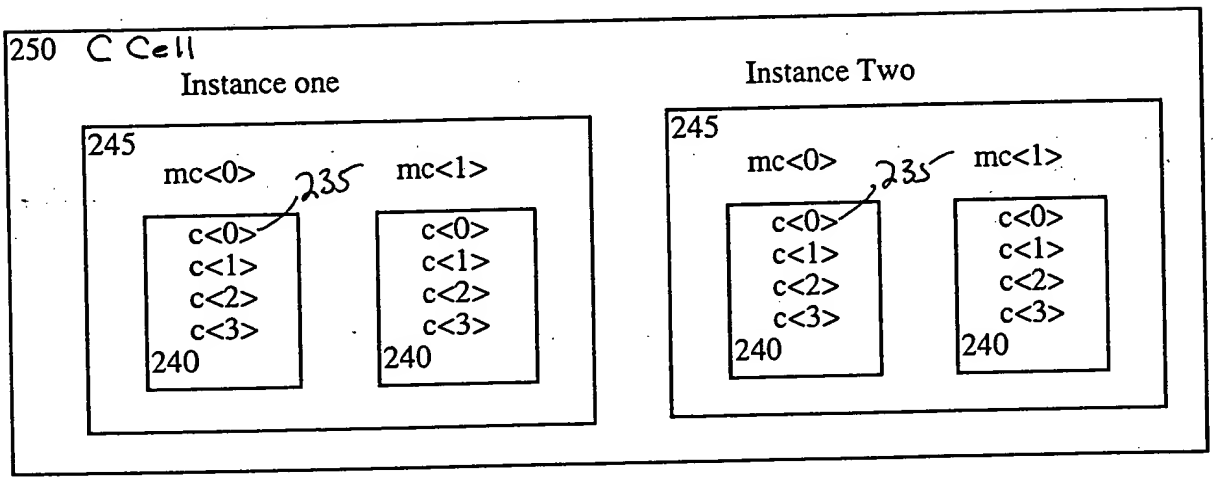

FIGURE 2D

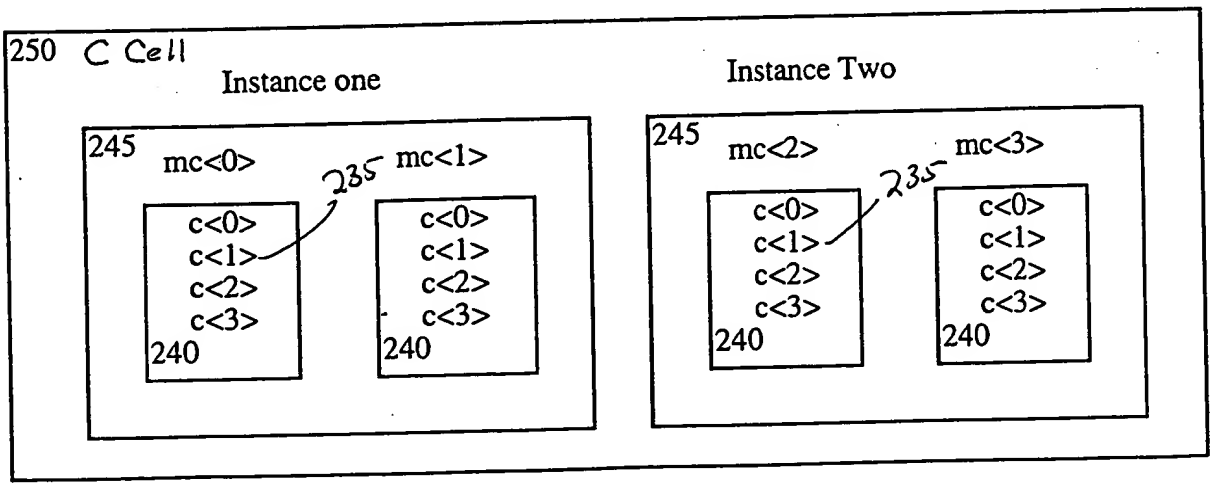

FIGURE 2E